**JUL 17**

Project Submission

Choose any area of the world in [https://www.openstreetmap.org](https://www.openstreetmap.org/) and use data munging techniques, such as assessing the quality of the data for validity, accuracy, completeness, consistency and uniformity, to clean the OpenStreetMap data for a part of the world that you care about. Choose to learn SQL or MongoDB and apply your chosen schema to the project.

**Evaluation**

A Udacity evaluator will review and check your completion of the problem sets in your choice of MongoDB or SQL, as well as your answers to the questions listed in the rubric.

Be sure to complete all of questions in the problem sets and the rubric before submitting your project.

**Rubric**

Your project will be evaluated by a Udacity reviewer according to this [Project Rubric](https://docs.google.com/document/d/1TpfNxDzUjhibq9Qb8cOQHtlvZUelft-W0fb7pCTTyYE/pub). Be sure to review it thoroughly before you submit. Your "project meets specifications" if it meets specifications in all the criteria

**Submission**

Ready to submit your project? Collect the following files:

* A pdf document containing your answers to the rubric questions. This file should document your data wrangling process.
* Your Python code for Lesson 6 quizzes, as well as any additional code you used in auditing and cleaning your dataset for the final project.
* A text file containing a link to the map position you wrangled in your project, a short description of the area and a reason for your choice.
* An .osm file containing a sample part of the map region you used (around 1 - 10 MB in size). See the notes below.
* A text file containing a list of Web sites, books, forums, blog posts, github repositories etc that you referred to or used in this submission (Add N/A if you did not use such resources).

Then go back to your Udacity Home, click on the project, and follow the instructions to submit:

* If you want to submit your files through a "Link to Project", upload your project files onto Github and send us the link.
* If you instead want to submit your files through "Upload a Zip", compress your project directory, and submit that zip file.

It can take us up to a week to grade the project, but in most cases it is much faster. You will get an email once your submission has been reviewed.

If you are having any problems submitting your project or wish to check on the status of your submission, please email us at dataanalyst-project@udacity.com.

## Project Details

This project is connected to the Data Wrangling course. You have the choice between two databases for this project: SQL and MongoDB. For an explanation of the differences between these two databases, see [this node](https://classroom.udacity.com/nanodegrees/nd002/parts/0021345404/modules/316820862075463/lessons/3168208620239847/concepts/77032306660923). There are separate instructions where relevant below for each database choice.

Here's what you should do:

### Step One - Complete Programming Exercises

Make sure all programming exercises are solved correctly in the "Case Study: OpenStreetMap Data" Lesson in the course you have chosen (MongoDB or SQL). This is the last lesson in that section.

### Step Two - Review the Rubric and Sample Project

The [Project Rubric](https://docs.google.com/document/d/1TpfNxDzUjhibq9Qb8cOQHtlvZUelft-W0fb7pCTTyYE/pub). will be used to evaluate your project. It will need to Meet Specifications for all the criteria listed. Here are examples of what your final report could look like:

[SQL Sample Project](https://gist.github.com/carlward/54ec1c91b62a5f911c42#file-sample_project-md)

[MongoDB Sample Project](https://docs.google.com/document/d/1F0Vs14oNEs2idFJR3C_OPxwS6L0HPliOii-QpbmrMo4/pub)

### Step Three - Choose Your Map Area

Choose any area of the world from [https://www.openstreetmap.org](https://www.openstreetmap.org/#map=5/51.500/-0.100), and download a XML OSM dataset. The dataset should be at least 50MB in size (uncompressed). We recommend using one of following methods of downloading a dataset:

* Download a preselected metro area from [Map Zen](https://mapzen.com/data/metro-extracts/).
* Use the [Overpass API](http://overpass-api.de/query_form.html) to download a custom square area. Explanation of the syntax can found in the [wiki](http://wiki.openstreetmap.org/wiki/Overpass_API). In general you will want to use the following query:(node(minimum\_latitude, minimum\_longitude, maximum\_latitude, maximum\_longitude);<;);out meta; e.g. (node(51.249,7.148,51.251,7.152);<;);out meta; the meta option is included so the elements contain timestamp and user information. You can use the Open Street Map[Export Tool](http://www.openstreetmap.org/export#map=5/42.618/-7.559) to find the coordinates of your bounding box. Note: You will not be able to use the Export Tool to actually download the data, the area required for this project is too large.

## Step Four - Process your Dataset

It is recommended that you start with the problem sets in your chosen course and modify them to suit your chosen data set. As you unravel the data, take note of problems encountered along the way as well as issues with the dataset. You are going to need these when you write your project report.

**Hint:** You may want to start out by looking at a smaller sample of your region first when auditing it to make it easier to iterate on your investigation. See code in the notes below for how to do this.

### SQL

Thoroughly audit and clean your dataset, converting it from XML to CSV format. Then import the cleaned .csv files into a SQL database using the schema provided or a custom schema of your choice.

### MongoDB

Thoroughly audit and clean your dataset, converting it from XML to JSON format. Then import the cleaned .json file into a MongoDB database.

## Step Five - Explore your Database

After building your local database you’ll explore your data by running queries. Make sure to document these queries and their results in the submission document described below. See the[Project Rubric](https://docs.google.com/document/d/1TpfNxDzUjhibq9Qb8cOQHtlvZUelft-W0fb7pCTTyYE/pub) for more information about query expectations.

## Step Six - Document your Work

Create a document (pdf, html) that directly addresses the following sections from the [Project Rubric](https://docs.google.com/document/d/1TpfNxDzUjhibq9Qb8cOQHtlvZUelft-W0fb7pCTTyYE/pub).

* Problems encountered in your map
* Overview of the Data
* Other ideas about the datasets

Try to include snippets of code and problematic tags (see [MongoDB Sample Project](https://docs.google.com/document/d/1F0Vs14oNEs2idFJR3C_OPxwS6L0HPliOii-QpbmrMo4/pub) or [SQL Sample Project](https://gist.github.com/carlward/54ec1c91b62a5f911c42#file-sample_project-md)) and visualizations in your report if they are applicable.

Use the following code to take a systematic sample of elements from your original OSM region. Try changing the value of k so that your resulting SAMPLE\_FILE ends up at different sizes. When starting out, try using a larger k, then move on to an intermediate k before processing your whole dataset.

*#!/usr/bin/env python*

*# -\*- coding: utf-8 -\*-*

**import** xml.etree.ElementTree **as** ET *# Use cElementTree or lxml if too slow*

OSM\_FILE = "some\_osm.osm" *# Replace this with your osm file*

SAMPLE\_FILE = "sample.osm"

k = 10 *# Parameter: take every k-th top level element*

**def** **get\_element**(osm\_file, tags=('node', 'way', 'relation')):

"""Yield element if it is the right type of tag

Reference:

http://stackoverflow.com/questions/3095434/inserting-newlines-in-xml-file-generated-via-xml-etree-elementtree-in-python

"""

context = iter(ET.iterparse(osm\_file, events=('start', 'end')))

\_, root = next(context)

**for** event, elem **in** context:

**if** event == 'end' **and** elem.tag **in** tags:

**yield** elem

root.clear()

**with** open(SAMPLE\_FILE, 'wb') **as** output:

output.write('<?xml version="1.0" encoding="UTF-8"?>\n')

output.write('<osm>\n ')

*# Write every kth top level element*

**for** i, element **in** enumerate(get\_element(OSM\_FILE)):

**if** i % k == 0:

output.write(ET.tostring(element, encoding='utf-8'))

output.write('</osm>')

### Reminder! To complete this project, you have the option of choosing either SQL or MongoDB.

If you need help deciding which option is best for you, consider the following:

Do you have previous experience with SQL? If so, consider learning MongoDB.

Do you have previous experience with MongoDB? If so, try SQL.

Don't have much experience in either one? We recommend starting with SQL.

SQL is more broadly adopted, giving you a great base to work from, and is more likely to be desired by potential employers.

Here are some additional resources that may help explain the difference and help you make your decision.

[SQL vs. NoSQL: The Differences](http://www.sitepoint.com/sql-vs-nosql-differences/)

[Should a Newbie Learn SQL or NoSQL?](https://www.quora.com/Should-a-newbie-learn-SQL-or-NoSQL)

[Should I Learn SQL or MongoDB and Why?](https://www.quora.com/Should-I-learn-SQL-or-NoSQL-MySQL-or-MongoDB-And-why)

### Checkout the supporting courses to make your final decision. You can change your mind anytime before you submit the project.

[MongoDB](https://classroom.udacity.com/nanodegrees/nd002/parts/0021345404/modules/316820862075462/lessons/745498943/concepts/7597386010923)

[SQL for Data Analysts](https://classroom.udacity.com/nanodegrees/nd002/parts/0021345404/modules/316820862075461/lessons/5391955257/concepts/53852961130923)